

PRAKTIKUM PEMROGRAMAN JARINGAN
JOBSHEET 7
“Version Control and App Deployment”



DOSEN PENGAMPU:
Randi Proska Sandra, M.Sc.

OLEH:
Muhammad Alfarobi
23343011

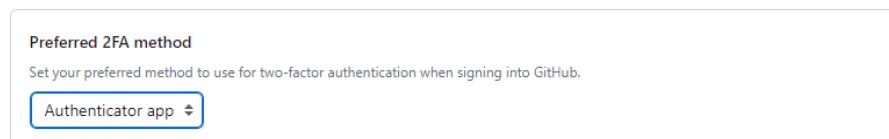
PROGRAM STUDI INFORMATIKA
DEPARTEMEN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2025

LATIHAN

A. Version Control Menggunakan Git

a. Registrasi Github

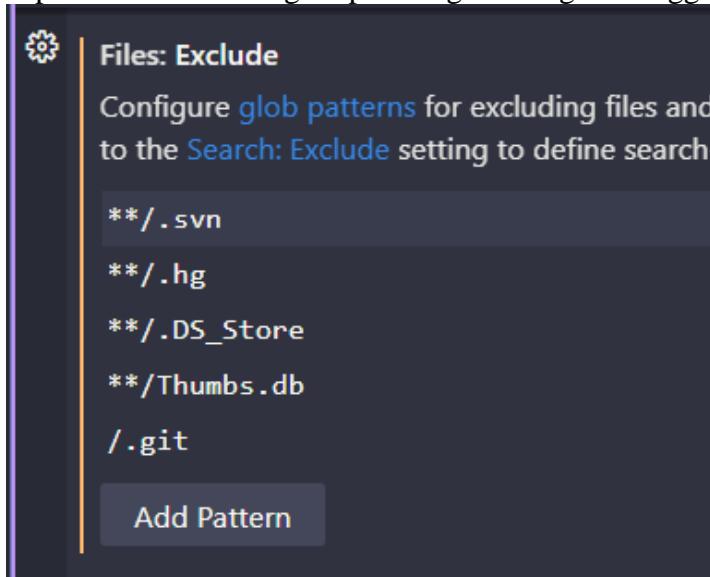
1. Silakan lakukan instalasi akun Github melalui link berikut ini <https://github.com/signup>. Disarankan menggunakan email institusi atau kampus seperti @student.unp.ac.id
2. Setelah pembuatan akun, silakan login. Disarankan juga untuk melakukan pengaturan Two-factor authentication. Anda dapat melakukannya dengan mengakses profile picture pada bagian kanan atas, lalu pilih menu Setting
3. Selanjutnya pilih menu Password and Authentication, lalu pilih bagian Two-factor authentication. Anda dapat memilih berbagai jenis metode autentikasi seperti gambar berikut
4. Disarankan memilih menggunakan **Authenticator app** atau **GitHub Mobile**. Opsi lainnya juga diperbolehkan. Perlu diperhatikan, jika anda memilih SMS/Tex message pastikan bahwa nomor handphone yang anda gunakan selalu aktif.
5. Jika anda memilih **authenticator app**, anda diharuskan mendownload aplikasi autentikasi seperti **Google Authenticator** dan **Microsoft Authenticator**. Silakan download aplikasi ini melalui PlayStore atau AppStore di smartphone anda. Setelah mendownload aplikasi autentikasi, silakan login aplikasi tersebut menggunakan akun gmail (Google) atau outlook (Microsoft).
6. Setelah itu, silakan scan barcode yang tampil di halaman GitHub menggunakan aplikasi autentikator yang anda pilih. Akan muncul kode angka dan silakan masukan kode ini di halaman GitHub. Lalu, selesaikan proses aktivasi autentikasi tersebut.
7. Jika anda mengaktifkan beberapa aplikasi autentikator, maka anda dapat memilih metode autentikasi apa yang diinginkan ketika anda login ke GitHub melalui bagian **Preferred 2FA method**



8. Autentikasi ini digunakan untuk melindungi akun anda sehingga setiap anda masuk anda akan diminta untuk memasukan kode autentikasi. Anda dapat mengakses kode autentikasi ini sesuai dengan jenis metode autentikasi yang anda pilih
- b. Pengaturan Git Commit untuk Version Control
 1. Silakan Download Git melalui link berikut ini <https://git-scm.com/>. Downloadlah sesuai sistem operasi anda
 2. Lakukan instalasi Git. Setelah selesai, silakan ketikan perintah berikut git --version melalui terminal di visual studio code untuk memastikan instalasi telah berhasil

```
C:\Users\HP>git --version
git version 2.37.3.windows.1
```

3. Selanjutnya silakan pastikan anda berada pada direktori web-server (aplikasi yang anda kembangkan pada modul sebelumnya) dan ketikan perintah berikut ini **git init** melalui terminal. Perhatikanlah bagian explorer di visual studio code, bahwa folder- folder pada direktori web-server anda berubah menjadi warna hijau
4. Ketika anda melakukan perintah **git init** akan keluar informasi yang menyatakan bahwa git repository telah di inisialisasi
5. Klik menu File – Preferences – Setting pada visual studio code anda dan akan muncul tampilan berikut ini
6. Drag ke bawah atau cari bagian File:Exclude seperti gambar dibawah ini, dan hapuslah tanda bintang ****** pada bagian ****/.git** sehingga menjadi **/.git**



7. Perhatikan pada bagian explorer, bahwa folder baru **.git** telah bertambah di direktori aplikasi web-server anda
8. Langkah tadi hanya untuk mengecek bahwa direktori tersebut telah ada dalam aplikasi anda. Silakan kembalikan lagi tanda ****** sehingga folder tersebut tersembunyi
9. Selanjutnya jalankan perintah **git status** untuk mengecek status tracking folder, akan keluar info terkait semua folder dan file berwarna merah
10. Pada proses version control ini, kita tidak akan memasukan folder dan file **node_modules** karena folder ini berisi library yang kita butuhkan dan bukan bagian utama dari source code aplikasi yang selalu kita edit. Buatlah file baru dalam folder web-server dan beri nama **.gitignore** (tanda titik . wajib ada).
11. Masukanlah teks berikut **node_modules/** pada file **.gitignore**
12. Jalankan lagi **git status** dan perhatikan bahwa sekarang direktori dan file yang belum di track hanya tersisa

```
.gitignore
package-
lock.json
package.jso
n public/
src/
templates/
```

13. Sedangkan folder **node_modules** akan berwarna abu-abu yang berarti folder tersebut akan dikecualikan
 14. Selanjutnya pada terminal, ketik perintah **git add src/**
 15. Jalankan lagi git status dan pastikan bahwa tidak ada direktori atau file yang belum di track (berwarna merah)
 16. Jika masih ada jalankan perintah berikut **git add .**
 17. Jalankan kembali git status dan pastikan bahwa tidak ada lagi yang berwarna merah atau belum ditrack.
 18. Selanjutnya ketikan perintah **git commit -m "Init commit"** untuk menyimpan perubahan yang telah dilakukan dan perhatikan bahwa semua direktori sekarang tidak berwarna hijau lagi
 19. Setelah proses ini, setiap kali anda mengedit baris kode pada file aplikasi anda, folder atau file tersebut akan berwarna orange yang mengindikasikan bahwa telah terjadi perubahan. Anda harus menjalankan perintah **git status**, **git add .** dan **git commit -m "Init commit"** kembali untuk melakukan sinkronisasi perubahan
 20. Ketikan juga perintah berikut ini **git commit -m "Remove unnecessary console.log call"** untuk menghapus beberapa baris console.log yang tidak penting
 21. Apabila semuanya telah selesai di commit menjadi proyek git dan tidak ada perubahan pada proyek aplikasi anda, maka setiap kali anda menjalankan perintah **git commit** akan keluar informasi berikut yang mengindikasikan bahwa tidak perubahan pada repository aplikasi anda
- c. Membuat SSH Key
1. Silakan buka terminal pada visual studio code dan pastikan anda berada pada direktori web-server
 2. Silakan ketik perintah berikut ini untuk menampilkan list konten pada direktori ssh
`Get-ChildItem -Force -File ~/.ssh`
 3. Selanjutnya masukan perintah berikut ini `ssh-keygen -t rsa -b 4096 -C randiproska@gmail.com` untuk membuat RSA SSH Key pair. Silakan ganti email dengan email anda.
 4. Hal ini akan meng-generate SSH key, biarkan semua nilai (valuenya) default dengan menekan enter hingga selesai atau tampil lebih kurang seperti gambar berikut
 5. Selanjutnya jalankan lagi perintah `Get-ChildItem -Force -File ~/.ssh` dan perhatikan bahwa sebuah RSA key telah di generate dengan dua nama file yaitu **id_rsa** dan **id_rsa.pub**
 6. RSA SSH Key ini akan digunakan oleh github nantinya. Kemudian jalankan perintah berikut ini satu persatu

```
$env:SSH_AUTH_SOCK = ""  
$env:SSH_AGENT_PID = ""  
Start-Process ssh-agent -Wait -NoNewWindow
```
 7. Kemudian jalankan perintah berikut ini `ssh-add 'C:\Users\Randi Proska Sandra\.ssh\id_rsa'` untuk menambahkan SSH private key ke SSH agent di Windows. Silakan sesuaikan direktori dengan direktori lokasi file id_rsa anda

sesuai gambar pada nomor 5

- d. Membuat Github Repository dan Push Aplikas ke Github
 1. Silakan login pada akun github anda dan buat repository baru dengan mengklik ikon tambah, lalu pilih new repository pada bagian kanan atas
 2. Silakan sesuai pengaturan seperti gambar berikut. Cukup ubah sesuai gambar (repository name, description, public), selebihnya biarkan default
 3. Lalu klik Create Repository dan akan tampil tampilan seperti berikut ini yang berisi informasi tentang command-command untuk setup repository
 4. Pada langkah selanjutnya kita akan memasukan aplikasi yang dibuat ke repository GitHub. Silakan masukan perintah berikut pada terminal di visual studio code **git remote add origin**
https://github.com/randiproska/aplikasiCuaca.git
Silakan sesuai nama akun dan nama aplikasi anda
 5. Selanjutnya buka kembali github, klik ikon gambar anda kanan atas dan pilih setting, lalu SSH dan GPG Keys. Kemudian, pilih New SSH Key
 6. Lalu pada tampilan berikut silakan diisi judul SSH key
 7. Silakan ketik perintah berikut **cat ~/.ssh/id_rsa.pub** pada terminal di visual studio code.
 8. Copy kan semua kode yang muncul pada langkah 7 ke bagian key seperti pada gambar di langkah 6. Pastikan bahwa tidak ada satu karakter pun tertinggal.
 9. Lakukan tes koneksi antara windows dan GitHub anda dengan memasukan perintah berikut pada terminal di visual studio code **ssh -T git@github.com**
 10. Lalu, tekan **enter** dan pilih yes
 11. Selanjutnya jalankan **git branch -M main** dan kemudian **git push -u origin main** seperti gambar berikut ini
 12. Perhatikanlah bahwa sekarang aplikasi anda telah terupload ke akun GitHub anda

```
$ git commit -m "add web-server to another repo for deploying"
[main (root-commit) d0c5a6f] add web-server to another repo for deploying
 28 files changed, 2388 insertions(+)
$ git push origin main
Enter passphrase for key '/c/Users/FLORE/.ssh/id_ed25519':
Enumerating objects: 39, done.
Counting objects: 100% (39/39), done.
Delta compression using up to 12 threads
Compressing objects: 100% (37/37), done.
Writing objects: 100% (39/39), 1.13 MiB | 3.37 MiB/s, done.
Total 39 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
```

B. App Deployment Menggunakan Cyclic

a. Registrasi Cyclic

Cyclic adalah platform cloud yang menyediakan cara mudah untuk menyebarluaskan (deploy) dan mengelola (manage) aplikasi yang dikembangkan oleh developer. Ikuti langkah-langkah berikut untuk registrasi

1. Silakan kunjungi halaman berikut ini <https://www.cyclic.sh/>, kemudian lakukan **sign up** menggunakan akun github dan setujui autentikasinya (jika diminta)
2. Setelah registrasi selesai akan tampil tampilan seperti berikut ini
3. Jika tampilan berbeda, silakan akses url berikut <https://app.cyclic.sh/>

b. Pengubahan File Sebelum Deplooyment

1. Silakan buka kembali direktori web-server anda melalui visual studio code.
2. Buka file package.json yang berada didalam direktori web-server dan gantilah kode bagian script menjadi seperti berikut ini
3. Untuk menguji bahwa aplikasi tetap berjalan normal, silakan jalankan perintah npm run start dan akses localhost:3000 pada web browser anda
4. Lalu buka file src/app.js, dan tambahkan baris kode berikut const port = process.env.PORT || 3000 seperti pada gambar
5. Pada kode app.listen bagian akhir, ubah menjadi seperti gambar berikut
6. Lalu buka lagi file public/js/app.js dan ubahlah url pada bagian fetch menjadi seperti gambar berikut ini
7. Jalankan kembali npm run start seperti langkah nomor 3 untuk menguji bahwa tidak ada masalah dengan program anda.
8. Jika telah selesai jalankan kembali perintah git dibawah ini satu persatu untuk mengupdate versi terbaru aplikasi anda ke github repository
9. git status
10. git add .
11. git commit -m "Init commit"
12. git push -u origin main

c. App Deployment

1. Pada gambar bagian a nomor 2, silakan pilih Link Your Own, lalu connectkan akun GitHub (jika diminta)
2. Jika akun telah terkoneksi maka akan muncul list repository anda seperti gambar berikut dan silakan pilih nama aplikasi yang telah di upload ke GitHub melalui proses version control
3. Nama repository pada modul ini adalah aplikasiCuaca, lalu pada tampilan selanjutnya silakan klik **Connect Cyclic**. Lalu tunggu proses selesai.
4. Setelah selesai, akan muncul informasi seperti gambar berikut
5. Silakan klik url tersebut dan website anda telah online melalui hosting cyclic

Dikarenakan Cyclic di shutdown, penulis menggunakan Vercel.

```
vercel.json > ...
{
  "version": 2,
  "builds": [
    { "src": "src/app.js", "use": "@vercel/node" }
  ],
  "routes": [
    { "src": "/(.*)", "dest": "src/app.js" }
  ]
}

// MEMULAI SERVER
// app.listen(port, () => {
//   console.log(`Server mulai ing http://localhost:${port}`);
// })

module.exports = app;
```

The screenshot shows the Vercel deployment interface. At the top, it says "Let's build something new". Below that is a search bar with placeholder text "Enter a Git repository URL to deploy...". To the right is a "Continue" button. The main area has two sections: "Import Git Repository" and "Clone Template". Under "Import Git Repository", there are fields for "Select a Git Namespace" and "Search...". A note says "Install the GitHub application for the accounts you wish to Import from to continue" and a "Install" button. Under "Clone Template", there are cards for "Next.js Boilerplate", "Vite + React Starter", and "Express.js on Vercel". Below these sections is a configuration for repository selection:

- All repositories: "This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only)."
- Only select repositories: "Select at least one repository. Also includes public repositories (read-only)."

A "Select repositories" button is shown, and below it, it says "Selected 1 repository."

All deployments

Latest deployments from pinned environments

A single deployment card for "github-pages". It shows a green checkmark icon and the text "github-pages". Below it says "Last deployed 5 minutes ago" and provides a deployment URL: https://robbi199.github.io/Aplikasi-Cek-Cuaca_Muhammad-Alfarobi/. There is also an "Open" button with a right-pointing arrow icon.

Pencarian Berita Cuaca

Masukkan kata kunci untuk mencari berita cuaca terkini:

Contoh: hujan, suhu, badai...

Cari Berita

Belum ada hasil pencarian.